Listing of Claims

Please amend the claims as follows:

1. [Cancelled]

2. [Previously Presented] The method of claim 34, wherein each of the data sequences has a length of one byte.

3. [Previously Presented] The method of claim 34, wherein each of the data sequences has a length of one bit.

4. [Previously Presented] The method of claim 34, wherein the instruction received from the processor is a member of an extensible instruction set.

5. [Previously Presented] The method of claim 34, wherein each of the one or more unaligned data sequences has a length less than a length of the second aligned word.

6-30. [Cancelled]

31. [Previously Presented] A method of processing data, the method comprising:

receiving a sequence of aligned data, the aligned data being aligned relative to a memory;

receiving a user defined instruction;

loading unaligned data which is a subset of the sequence of aligned data into a programmable instruction set extension fabric;

executing the user defined instruction using the unaligned data to create an unaligned instruction output;

receiving the unaligned instruction output;

aligning unaligned instruction output to the memory to create aligned data; and

storing the aligned data in the memory.

32. [Previously Presented] The method of claim 31, wherein receiving the sequence of aligned data is controlled by a processor using a load/store instruction, and the user defined instruction is received from the processor.

33. [Previously Presented] The method of claim 31, wherein the sequence of aligned data is received from a first location within the memory and the aligned data is written to a second different location within the memory.

34. [Currently Amended] A method for processing data sequences in a computing system, the method comprising:

initializing a load/store buffer in an extension adapter by loading a first aligned word into a load/store buffer;

further initializing the load/store buffer by loading a second aligned word into the load/store buffer, alignment of the first aligned word and the second aligned word being relative to a memory accessible via a processor;

reading an unaligned data sequence from the load/store buffer into a register file of an extension adapter for use by an instruction received from the processor, the unaligned data sequence including at least part of the second aligned word;

loading additional aligned words to the load/store buffer to replace the first aligned word and the second aligned word; ~~and~~

executing the unaligned data sequence; and

changing a memory address pointer by an amount less than a length of the second aligned word to point to a next unaligned data sequence to be read.

35-38. [Canceled]